P.A. 2

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-051861
(43)Date of publication of application : 23.02.2001

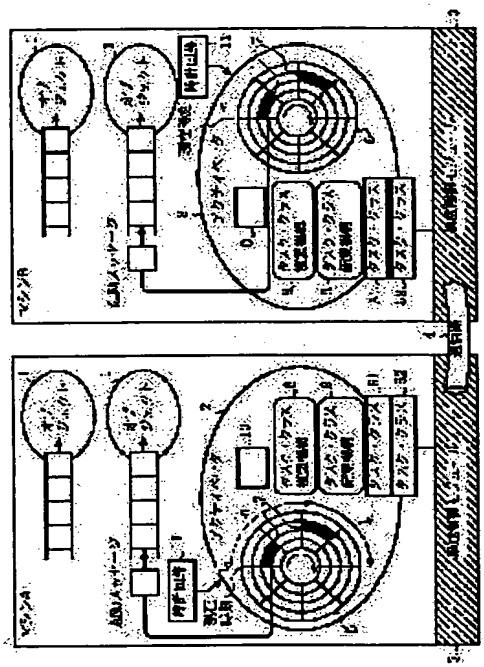
(51)Int.CI. G06F 9/46
G06F 11/20
G06F 15/16

(21)Application number : 11-226614
(22)Date of filing : 10.08.1999

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : SHIMAKAWA HIROMITSU

(54) DOUBLED SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To obtain a doubled system wherein a slave-system machine is placed in a standby state and immediately replaces a master-system machine if the master-system machine becomes abnormal by copying task classes from the master-system machine in a normal state to the slave-system machine in a cold standby state.
SOLUTION: This doubled system holds task classes 51, 52... in a memory so that they can dynamically be corrected, enables an activator 2 to make a mode shift, and provides a mode wherein the task classes are copied as one of modes, and is equipped with a task class copying mechanism 8 which copies the task classes in the mentioned copy mode and a task class editing mechanism 9 which edits the task classes at execution time.



LEGAL STATUS

[Date of request for examination] 21.10.2003
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

P.A.2.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－51861
（P2001－51861A）
(43)公開日　平成13年２月23日(2001.2.23)

(51)Int.Cl.⁷	識別記号	FI	テーマコード（参考）
G06F 9/46	360	G06F 9/46	360G　5B034
11/20	310	11/20	310C　5B045
15/16	640	15/16	640K　5B098

審査請求　未請求　請求項の数２　ＯＬ　（全 13 頁）

(21)出願番号　特願平11－226614

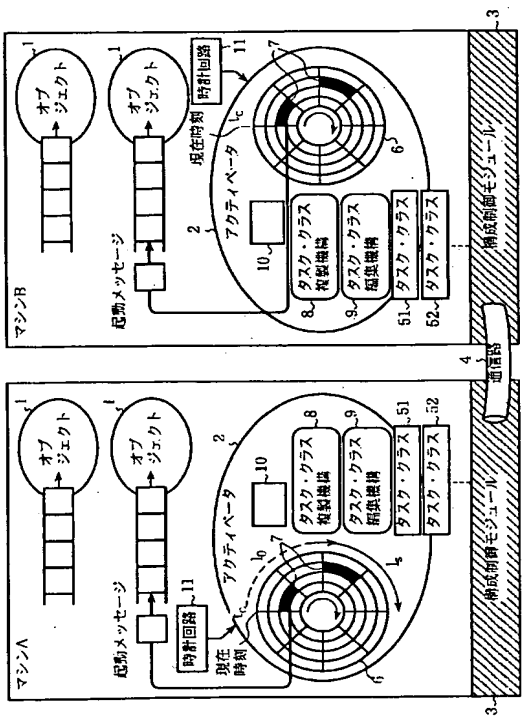
(22)出願日　平成11年８月10日(1999.8.10)

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号
(72)発明者　島川　博光
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内
(74)代理人　100066474
弁理士　田澤　博昭　（外１名）

Ｆターム(参考)　5B034 BB02 BB17 CC02 DD06 DD07
5B045 JJ24 JJ26 JJ44
5B098 AA10 DD01 FF01 GA04 GB01

(54)【発明の名称】　二重化システム

(57)【要約】
【課題】　コールド・スタンバイ状態の従系マシンに正常状態の主系マシンからタスク・クラスを複写することによって、従系マシンをホット・スタンバイ状態にしておき、主系マシンに異常が発生したときは従系マシンが即座に入れ替われる二重化システムを得る。
【解決手段】　タスク・クラス５１，５２，…をメモリ上に動的に修正可能に保持し、アクティベータ２もモード遷移可能として、そのモードの１つとしてタスク・クラスを複製するモードをもたせるとともに、このタスク・クラスを複製するモードにおいてタスク・クラスの複製を行うタスク・クラス複製機構８と、実行時にタスク・クラスの編集を行うタスク・クラス編集機構９を設けた。



1

【特許請求の範囲】
【請求項１】　一方が主系、他方が従系となる２つのマシンを用い、主系のマシンの異常時に従系のマシンがそのサービスを引き継ぐ二重化システムにおいて、
タスクを実行し、かつ、モード遷移可能な、１つ以上のオブジェクトと、
メモリ上に動的に修正可能に保持された、実行すべき前記タスクの集合を示すタスク・クラスと、
前記タスク・クラスから作成したタスク・スケジュールをもとに、前記オブジェクトにタスク実行のタイミングを知らせ、かつ、モード遷移可能で、そのモードの１つとして、マシン間でタスク・クラスの複製を行うモードを有するアクティベータと、
前記アクティベータの複製を行うモードにて動作し、マシン間でのタスク・クラスの複製を行うタスク・クラス複製機構と、
実行時に前記タスク・クラスの編集を行うタスク・クラス編集機構と、
自らのマシンでの異常検知もしくは相手マシンとの通信によって、自らのマシンの遷移すべきモードを決定する構成制御モジュールとを、前記マシンのそれぞれにもたせ、
それら２つのマシンを、互いの稼動状態を通知し合うための通信路で接続したことを特徴とする二重化システム。
【請求項２】　各マシンが、２つのマシン間で同期された時計回路をもつことを特徴とする請求項１記載の二重化システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、周期的に、もしくは予約された時刻に、タスクを実行するデータ処理システムの機能を、システム内のトラブルとは無関係に、ユーザからみて連続して使えるようにする二重化システムに関するものである。
【０００２】
【従来の技術】データ処理システムを構成する計算機などのマシンにおいて、例えばディスクに異常があり、計算の途中でディスクへのデータ書き込みが失敗したとする。この場合、当該マシンではそれ以上作業の続行が困難になることがある。そのような場合には、システムをマシンＡとマシンＢとで二重化しておき、例えばマシンＡでそれ以上作業の続行が困難になると、マシンＢに対して作業の続行を依頼する。処理結果に高い信頼性が要求されるシステムにおいては、このように、たとえ一方のマシンにトラブルがあったとしても、他方のマシンが引き続き同様の作業を行うようにして、ユーザからみてあたかも連続して１つのシステムが使えるようにしなければならない。
【０００３】このようなマシンＡおよびマシンＢの２つ

2

のマシンで構成される二重化システムにおいては、一方のマシンが主系、他方のマシンが従系となり、通常は主系のマシンだけが動作して、従系のマシンは待機状態にある。これは主系マシンが作業している間に従系マシンが作業することによって作業の重複などの矛盾が生じてはいけないからである。主系となっているマシンにトラブルが発生してそれ以上作業の続行ができなくなった場合、今まで従系となっていたマシンが主系となって作業実行を継続する。
【０００４】従来の二重化システムは、一方のマシンが異常事態に陥ったときに、他方のマシンを起動してサービスを引き継ぐコールド・スタンバイ方式と、常時２つのマシンが同じタスクを実行していて、一方のマシンが異常事態に陥ったときに、他方のマシンが即座にそのサービスを引き継ぐホット・スタンバイ方式の２つに大別される。
【０００５】図５は例えば、水沼一郎、神余浩夫、島川博光、竹垣盛一による「実時間制約を保証する待機冗長システムの一設計手法」（電子情報通信学会論文誌、Ｖｏｌ．Ｊ７８－Ｄ－Ｉ，Ｎｏ．８、１９９５．８）に示された、従来のコールド・スタンバイ方式による二重化システムを示す構成図である。図において、１はオブジェクト、２はアクティベータ、３は構成制御モジュール、４は通信路、５はタスク定義ファイルである。
【０００６】また、図６はそのアクティベータ２の構成を示す説明図であり、相当部分には図５と同一符号を付してその説明を省略する。図において、６は当該アクティベータ２内のスケジュールテーブル、７はそのスケジュールテーブル６内のタスク・インスタンスであり、５１，５２，…はタスク定義ファイル５内のタスク・クラスである。
【０００７】次に動作について説明する。まず、従来のコールド・スタンバイ方式の二重化システムにおけるサービス引き継ぎについて説明する。なお、この場合、マシンＡを主系のマシンとし、マシンＢを従系のマシンとする。いま、従系のマシンＢの構成制御モジュール３が、主系のマシンＡからの引き継ぎ通知を受け取るか、あるいは主系のマシンＡからの生存メッセージが受け取れなくなると、主系のマシンＡでの作業続行が不可能であると判断し、自らのマシン（マシンＢ）上のオブジェクト１を使ってサービス提供を開始する。このサービス引き継ぎの開始は、マシンＢ上のタスク定義ファイル５の内容を読んで、スケジュールを周期的に作成するアクティベータ２を起動することによって実現される。
【０００８】次に、従来のホット・スタンバイ方式の二重化システムでのサービス引き継ぎについて説明する。ホット・スタンバイ方式は主系のマシンとまったく同一の動作を従系のマシンに常に実行させておき、主系のマシンに異常が起こったときに、直ちに従系のマシンがそのサービスを引き継げるようにしたものである。このホ

ット・スタンバイ方式では、主系マシンと従系マシンは同時に起動され、それらのオブジェクト１、およびアクティベータ２は全く同じ動作をとるようにする。また、タスク定義ファイル５内のタスク・クラス５１，５２，…の修正も主系マシンと従系マシンとでそれぞれ同時に行われる。

【０００９】このようなホット・スタンバイ方式の二重化システムでは、主系のマシンと従系のマシンが同時に稼働しているが、これらがともに出力データを送出すると、同じデータが重複して出力対象に届くことになり、混乱を招くことがある。そのため、オブジェクト１はモードを持つことができ、従系のマシンは、同じメッセージを受け取っても、主系のマシンと同じ計算を行うが、データの出力は行わないようにする。

【００１０】また、コールド・スタンバイ方式の二重化システムでは、サービスの引き継ぎ時に新たに従系マシンが主系マシンとして動作を開始するため、ファイルの読み出し、メモリの初期化などの処置を行う必要がある場合が多い。このため、オブジェクト１はモードを遷移する際に、あらかじめ指定した手続きを実行できるようにしておくのが普通である。

【００１１】

【発明が解決しようとする課題】従来のコールド・スタンバイ方式の二重化システムは以上のように構成されているので、従系のマシンが新たに主系のマシンになろうとする際には、タスク・クラス５１，５２，…がタスク定義ファイル５より読み出され、実行時にタスク・クラス５１，５２，…がそれまで主系であったマシンで修正されている場合には対処できないという課題があった。

【００１２】なお、このように、従系マシンが新たに主系マシンになろうとする際に、タスク定義ファイル５よりタスク・クラス５１，５２，…を読み出す方式を採用している限り、タスク・クラス５１，５２，…の動的修正に対応するためには、主系マシンと従系マシンにおけるタスク定義ファイル５を、人手などの方法で、常に内容が一致するように維持しておくことが必要であり、また、この内容を一致させるための処理を行っても、従来のコールド・スタンバイ方式の二重化システムでは、主系マシンの異常が発見されてから、従系マシンでアクティベータ２が起動されるため、主系マシンでの異常発生から従系マシンでのサービス開始までに時間がかかるといった課題もあった。

【００１３】一方、従来のホット・スタンバイ方式の二重化システムでは、主系のマシンと従系のマシンはまったく同じ動作を常にとっていなければならないので、タスクが動的に変更され、不休のサービスが要求されるようなシステムにおいては、主系マシンに異常が起こったために新たに主系となったマシンに別の異常が起こったときには、これをバックアップしているマシンがないので対処できないという課題があった。

【００１４】この発明は、上記のような課題を解決するためになされたもので、タスク・クラスをメモリ上に保持し、これを動的に修正することが可能で、コールド・スタンバイ状態の従系のマシンに正常状態の主系マシンからタスク・クラスを複写することによって、従系のマシンをホット・スタンバイ状態にしておき、主系のマシンに異常が発生したときは従系のマシンが即座に入れ替われるようにした二重化システムを得ることを目的とする。

【００１５】

【課題を解決するための手段】この発明に係る二重化システムは、通信路で２つのマシンを接続し、それら各マシンにおいて、実行すべきタスクの集合を示すタスク・クラスを、メモリ上に動的に修正可能に保持し、そのタスク・クラスから作成したタスク・スケジュールをもとに、オブジェクトにタスクの実行タイミングを知らせるアクティベータをモード遷移可能とし、そのモードの１つとして、タスク・クラスの複製を行うモードを持たせるとともに、２つのマシンのそれぞれに、アクティベータにおけるタスク・クラスの複製を行うモードにおいて、タスク・クラスの複製を行うタスク・クラス複製機構と、実行時にタスク・クラスの編集を行うタスク・クラス編集機構とを設けたものである。

【００１６】この発明に係る二重化システムは、２つのマシン間で同期された時計回路を各マシンに持たせたものである。

【００１７】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態１．図１はこの発明の実施の形態１による二重化システムを示す構成図である。図において、１はタスクを実行する、少なくとも１つのオブジェクトであり、このオブジェクト１はモード遷移が可能となっている。２はこのオブジェクト１にタスク実行の起動タイミングを知らせるアクティベータであり、このアクティベータ２はモードが遷移可能となっていて、そのモードの１つとして、マシン間でタスク・クラスの複製を行うモードを有している点で、従来のそれとは異なっている。３は自らのマシンでの異常通知に対してその処置を司り、また相手のマシンが正常に稼働しているかを常時検査している構成制御モジュールである。４はこの構成制御モジュール３が互いのマシンの稼働状態を通知し合うための通信路である。

【００１８】また、５１，５２，…はオブジェクト１で実行されるタスクの集合を示すタスク・クラスであり、この場合、図示を省略したタスク定義ファイルなどに登録されているものが、マシンの立ち上げ時にメモリ上に動的に修正可能に保持されるものである点で、従来のそれとは異なっている。６はアクティベータ２内にあって、タスクの実行タイミングを示すスケジュールテーブ

ルであり、７はこのスケジュールテーブル６上に登録されているタスク・インスタンスである。８はアクティベータ２が有するマシン間でタスク・クラスを複製するモードにおいて動作し、マシン間でのタスク・クラス５１，５２，…の複製を行うタスク・クラス複製機構である。９は実行時にタスク・クラス５１，５２，…の編集を行うタスク・クラス編集機構であり、１０はこのタスク・クラス５１，５２，…の編集を禁止する際に有効となるタスク・クラス編集禁止フラグである。１１は現在時刻の情報を生成する時計回路であり、２つのマシン間で同期化されている。

【００１９】この二重化システムは、これらによって同一に形成されたマシンＡおよびマシンＢの２つのマシンによって構成されている。

【００２０】ここで、各構成要素の働きについて説明する。まず、オブジェクト１について説明する。オブジェクト１は途中の計算状態を保持するための内部メモリと計算方法を示したメソッドより構成されており、外部から起動メッセージを受け取ったとき、その起動メッセージにて指定されたメソッドを起動する。起動されたメソッドは内部メモリを使って計算を行う。メソッドの実行が終ればオブジェクト１は新たな起動メッセージが自らに届いていないかを確かめ、もし届いていれば同じようにメソッドを起動する。もし届いていなければ次の起動メッセージが届くまで休眠する。オブジェクト１はモードを持つことが許されており、同じメッセージを受け取ってもモードによっては、異なる動作を行うことがある。また、モードを遷移する際には指定されたメソッドを実行するように指定することもできる。なお、これについては後述する。

【００２１】次に、タスク・クラス５１，５２，…について説明する。タスク・クラス５１，５２，…はそれぞれ実行すべきタスクの集合を示している。なお、主系のマシンと従系のマシンでは、このタスク・クラス５１，５２，…の内容は通常、同一である。ここで、タスクには周期タスクと起動時刻指定タスクとがある。周期タスクでは、実行すべきオブジェクト１の識別子、オブジェクト１に送るべきメッセージ、起動周期などが指定され、起動時刻指定タスクでは、実行すべきオブジェクト１の識別子、オブジェクト１に送るべきメッセージ、起動時刻などが指定される。

【００２２】なお、上記起動時刻指定タスクの起動時刻は、複数ある場合には、起動時刻のリストとして指定されてもよいし、何らかの代数式表現を用いて起動時刻の集合として指定されてもよい。何らかの代数式表現を用いて起動時刻の集合を指定する後者の例として、例えば次式のようなものが考えられる。

｛ｃ｜ｃ　∈　ＷＯＲＫＩＮＧＤＡＹ　∧　ｃ＝ＮＯＯＮ｝

これは休日を除いた作業日の正午をあらわしており、この時刻にタスクが起動されることを示している。

【００２３】これら周期タスクにしても起動時刻指定タスクにしても、実際に起動されるタスクの集合を表していることになる。そこで、タスク・インスタンス（実際に起動される個々のタスク）７と、タスク・クラス（起動条件の指定により表現されるタスクの集合）５１，５２，…として２つを区別する。

【００２４】次に、アクティベータ２について説明する。アクティベータ２はタスク・クラス５１，５２，…の内容に応じて、周期的に近未来において起動すべきタスク・インスタンス７を決定し、起動メッセージを各オブジェクト１に送信してタスク・インスタンス７を起動する。これを実現するために、アクティベータ２はスケジュール・テーブル６を備えている。このスケジュール・テーブル６は、現在時刻をt_cとして、次式で示す期間l_wに起動すべきタスク・インスタンス７のスケジュールを保持しているものとする。

l_w：［t_c，t_c+w）

【００２５】アクティベータ２は周期的にタスク・クラス５１，５２，…を読み出し、次式で示す期間l_sにスケジュールすべきタスク・インスタンス７を決定し、それをスケジュール・テーブル６に書き込む。

l_s：［t_c+l_o，$t_c+l_o+l_s$）

【００２６】ここで、l_o，l_sをそれぞれ、スケジュール・オフセット、スケジュール幅と呼ぶことにする。なお、スケジュール・オフセットl_oはスケジュールすべき近未来の期間の始点とスケジュール作業開始時刻との差分であり、スケジュール幅l_sはスケジュールすべき近未来の期間の長さである。

【００２７】アクティベータ２はスケジュールを終了すると、現在時刻に起動時間が設定されているタスク・インスタンス７がスケジュール・テーブル６上に登録されているかどうかを調べる。もし登録されていれば、そのタスク・インスタンス７で指定されたオブジェクト１に、指定された起動メッセージを送信してタスクを起動する。

【００２８】ここで、主系のマシンに異常が発生したときに、即座に従系のマシンがそのサービスを引き継げ、かつ、動的にタスク・クラス５１，５２，…が修正される主系マシンのサービスを、一旦停止した従系マシンがいつでも引き継げるようにするために、この実施の形態では、主系のタスク・クラス５１，５２，…を従系のマシンに複製する機能をアクティベータ２に持たせている。すなわち、アクティベータ２はタスク・クラス複製機構８を備えており、従系マシンのアクティベータ２ではそれを用いて、主系マシン上のタスク・クラス５１，５２，…を従系マシン上に複製する。また、タスク・クラス編集機構９およびタスク・クラス編集禁止フラグ１０も備えており、タスク・クラス編集機構９にて、タスク・クラス５１，５２，…の追加・削除・修正などの編

集を動的に行う。なお、上記タスク・クラス複製機構８によるタスク・クラス５１，５２，…の複製時には、タスク・クラス編集禁止フラグ１０を有効としてタスク・クラス編集機構９による編集を禁止し、複製した従系マシン上のタスク・クラス５１，５２，…に矛盾が生じないようにしている。

【００２９】次に、構成制御モジュール３について説明する。構成制御モジュール３は各マシンに１つあり、それぞれのマシンのＩＯ装置などに異常があった場合の対処を管理するものである。この構成制御モジュール３の働きは、自らのマシン上での異常を相手マシン上の構成制御モジュール３に伝える自マシン異常の通知、および相手マシンが音信不通になったときに相手マシンに異常があったと判断する相手マシン異常の検知の２つに大別される。なお、異常でなくとも、保守などのために主系のマシンをユーザが意図的に従系のマシンに切替える場合は、ユーザからの系切替え信号を疑似的な異常状態の発生とみなし、マシン異常の中の特別な場合として扱う。

【００３０】自マシン異常の場合の例として、マシンＡのディスクに異常があり、マシンＡのどれかのオブジェクト１がディスクへのデータ書き込みに失敗したとし、マシンＡではこれ以上作業の続行が困難である場合を考える。この場合には、マシンＢにサービスの引き継ぎを依頼しなければならない。マシンＡ上でディスクへのデータ書き込みに失敗したオブジェクト１はその旨を自らのマシンの構成制御モジュール３に通知する。通知を受けた構成制御モジュール３は、相手マシン、すなわち、マシンＢの構成制御モジュール３に自ら（マシンＡ）がこれ以上作業を続けることができないことを通信路４を使って通知する。この通知を「引き継ぎ通知」と呼ぶことにする。引き継ぎ通知を受け取ったマシンＢ上の構成制御モジュール３は、マシンＢ上のオブジェクト１を使ってタスクを実行することを開始する。

【００３１】次に、相手マシンの異常検知とその対処について説明する。相手マシンの異常検知は音信不通になったことにより検知される。双方の異常を検知するため、構成制御モジュール３は通信路４を介して定期的に相手マシンの構成制御モジュール３に「生存メッセージ」を送出している。この生存メッセージが定期的に届いている限り、相手マシンは正常に動作していると判断できる。ところが、相手マシンから生存メッセージが届かなくなった場合、音信不通となったマシンは動作不能になったことすら伝えられない程の異常状態に陥っていると考えられる。主系マシンから生存メッセージが届かなくなった場合、従系マシンは主系マシンは既に動作不能であると判断して、引き継ぎ通知を待たずに自らのマシン上のオブジェクト１を使ってタスクを実行することを開始する。

【００３２】また、通信路４は前述のように、２つのマシンが互いの異常を通知したり、検知したりするために使われるもので、直結ラインやネットワークなどにより実現される。

【００３３】次に動作について説明する。ここで、主系マシンと従系マシンの動作の概要を図２に示す。この図２に図示の例では、マシンＡが主系でマシンＢが従系であるものとする。

【００３４】まず、従系のマシンＢは保守等の要因により、一旦、停止されていたものとする。この間、主系のマシンＡは、待機している従系マシンがなく「主系のみでの動作状態」にある。その後、従系のマシンＢが立ち上がる。このとき、従系のマシンＢ内には、いかなるタスクをスケジュールすべきかを示したタスク・クラス５１，５２，…を持っていない。従系であるマシンＢは主系のマシンＡのサービス引き継ぎができるようになるために、通信路４を介してタスク・クラス５１，５２，…の複製をマシンＡに依頼して「タスク・クラスを複製している状態」に移行する。この依頼を受け付けた主系のマシンＡは「タスク・クラスを複製されている状態」に移行する。

【００３５】主系のマシンＡが「タスク・クラスを複製されている状態」になり、従系のマシンＢが「タスク・クラスを複製している状態」になると、マシンＡのタスク・クラス５１，５２，…の集合が従系のマシンＢに複製される。このタスク・クラス５１，５２，…の複製は、それぞれのマシン上のタスク・クラス複製機構８を用いて行われる。そのとき、このタスク・クラス５１，５２，…と合わせて、主系のマシンＡから従系のマシンＢにスケジュール開始時刻に関する情報も通知される。スケジュールは時刻ｂから周期ｃごとに生成されてきたとすると、ｎを整数として、時刻（ｂ＋ｎ・ｃ）に主系のマシンＡ上でスケジュールされる。このスケジュール開始時刻に関する情報として、これら時刻ｂと周期ｃがマシンＡからマシンＢに通知されれば、タスク・クラスの複製が時刻 t_f に終了したとして、次式を満たす時刻 t_s がマシンＡとマシンＢの双方で同じスケジュールの作成を開始する時刻となる。

$$t_s = b + n_s \cdot c > t_f$$

【００３６】この場合、たとえ、主系のマシンＡの時計回路１１と従系のマシンＢとの間で時計回路１１が同期していなくても、整数 n_s の値をそれぞれのマシン上で適切に選ぶことにより、最悪で周期ｃだけずれたタイミングでスケジュール作成が開始される。

【００３７】なお、この主系のマシンＡの「タスク・クラスを複製されている状態」と従系のマシンＢの「タスク・クラスを複製している状態」では、タスク・クラス編集禁止フラグ１０を有効とすることで、タスク・クラス５１，５２，…の編集を禁止する。これは、そのときタスク・クラス５１，５２，…に追加や削除、修正などがあると、複製されたタスク・クラスに矛盾が生じるの

で、それを防止するためにタスク・クラス５１，５２，…の編集を禁止している。

【００３８】タスク・クラス複製機構８によるタスク・クラス５１，５２，…の複製が終了するとマシンＡとマシンＢとでタスク・クラス５１，５２，…の状態は等価化され、マシンＢはマシンＡに等価化終了を通信路４を用いて知らせる。これにより、主系のマシンＡは「待機している従系を持った状態」になり、従系のマシンＢは「待機している状態」になる。これらの状態では、マシンＡ、マシンＢはともに内部計算としては同じ計算処理を行う。その時、外部に対して重複した出力がないようにするため、従系のマシンＢでは計算結果の出力だけが抑制されるなどの動作が取られる。

【００３９】主系のマシンＡの「待機している従系を持った状態」と従系のマシンＢの「待機している状態」においては、近未来での期間のタスクをスケジュールし、その期間が訪れれば、タスクがスケジュールにしたがって起動される。例えば、期間[ｂ＋ｎ・ｃ，ｂ＋（ｎ＋１）・ｃ）で実行されるタスクのスケジュールの作成が時刻（ｂ＋ｎ・ｃ）にて開始され、ここで作られたスケジュールにしたがって先の期間のタスクが起動される。

【００４０】いま、主系のマシンＡでトラブルが発生して、マシンＡによるサービスの続行が不可能になったとする。これは主系のマシンＡあるいは従系のマシンＢのいずれかの構成制御モジュール３によって検知される。主系のマシンＡの構成制御モジュール３で検知された場合には、主系のマシンＡから従系のマシンＢに系切替え通知が送られる。また、従系のマシンＢの構成制御モジュール３で検知された場合には、マシンＢが自らに対して系切替え通知を出す。系切替え通知を受け取ったマシンＢは、マシンＡと同じスケジュールをもっているので、直ちにマシンＡのサービスを引き継ぐことができる。このときマシンＢは主系マシンとなって動作するが、その状態は「主系のみでの動作状態」である。

【００４１】メモリ上に現在登録されているタスク・クラス５１，５２，…の集合は、例えば、タスク・クラス５１，５２，…を表す要素のリストとして構成される。タスク・クラス編集機構９はこのリストに対して動的に、すなわち実行時に、現在のタスク・クラス５１，５２，…の集合に対して、新しいタスク・クラスの追加や、登録済のタスク・クラスの削除などのタスク・クラスの修正を行う。例えば、タスク・クラスのリスト表現への新しいタスク・クラスの追加はリストへの要素追加で実現でき、登録済のタスク・クラスの削除はリストからの要素の探索と削除により実現できる。ただし、タスク・クラス５１，５２，…の編集が許されているのは、タスク・クラス編集禁止フラグ１０が無効である場合のみである。もし、タスク・クラス編集禁止フラグ１０が有効である場合には、タスク・クラス５１，５２，…の編集は失敗するか、延期される。

【００４２】タスク・クラス複製機構８は、主系側と従系側で使われるそれぞれのメソッドを持っている。タスク・クラス複製機構８は、まずタスク・クラス編集禁止フラグ１０を有効にし、次いで自らが主系であるか従系であるかに応じて、主系側あるいは従系側の各々のメソッドを実行し、その後、タスク・クラス編集禁止フラグ１０を無効にする。主系側のメソッドでは、登録されているタスク・クラス５１，５２，…を従系側に通知し、従系側のメソッドでは送られてきたタスク・クラスを登録する。主系側のタスク・クラス複製機構８のメソッドは、例えば、タスク・クラス５１，５２，…を表す要素のリストを１つ１つ読み出し、これを通信路４を用いて従系側に転送することにより実現される。また、従系側のタスク・クラス複製機構８のメソッドは、例えば、送られてきたタスク・クラスを１つ１つタスク・クラス５１，５２，…のリスト表現に追加して行くことによって実現される。

【００４３】ここで、このアクティベータ２はタスク・クラス５１，５２，…を複製するために、いくつか（この場合には７種）のモードを遷移する。以下にそのモードを列挙する。なお、説明を簡単にするために、ここでもマシンＡを主系とし、マシンＢを従系とする。

【００４４】S_f ：マシンが立ち上がっていない、ストップ（Ｓｔｏｐ）モード

S_i ：マシンが立ち上がり、タスク・クラス５１，５２，…はまだ登録されていない、イニシャル（Ｉｎｉｔｉａｌ）モード

S_p ：従系のマシンＢが主系のマシンＡのタスク・クラスを複写している過渡モードである、プリペアリング（Ｐｒｅｐａｒｉｎｇ）モード

S_b ：従系のマシンＢ上でタスク・クラス５１，５２，…の複製が終了したモードで、自らが主系のマシンＡといつでも交替できるスタンバイ（Ｓｔａｎｄｂｙ）モード

S_o ：主系のみでの動作をする、マスタ・ウィズアウト・スタンバイ（Ｍａｓｔｅｒ－ｗｉｔｈｏｕｔ－ｓｔａｎｄｂｙ）モード

S_e ：主系のマシンＡが従系のマシンＢによってタスク・クラスを複写されている過渡モードである、イコライズド・マスタ（Ｅｑｕａｌｉｚｅｄ－ｍａｓｔｅｒ）モード

S_m ：主系のマシンのモードで、自らが稼働していて、かつ、従系のマシンがバックアップしている、マスタ・ウィズ・スタンバイ（Ｍａｓｔｅｒ－ｗｉｔｈ－ｓｔａｎｄｂｙ）モード

【００４５】なお、アクティベータ２は過渡モードであるプリペアリングモードS_p あるいはイコライズド・マスタモードS_e でタスク・クラス５１，５２，…の複製が終了すればつぎのモードに自動的に遷移する。

【００４６】次に、このアクティベータ２のモード遷移

について詳細に説明する。ここで、図３は１つのマシンのモード遷移を示す説明図である。最初にマシンＡを主系、マシンＢを従系として稼働開始させ、二重化システムがどのように振舞うかを例示する。

【００４７】二重化システムを稼働させるために、まず、マシンＡとマシンＢとを立ち上げる。このときのモードはイニシャルモードS_1である。次いでマシンＡだけをバックアップのない主系単体のマスタ・ウィズアウト・スタンバイモードS_0に遷移させる。ここでは、例えば、図示を省略したタスク定義ファイルなどに保持されているタスク・クラス５１，５２，…がメモリ上に動的に編集可能に登録される。以後、マシンＡはバックアップするマシンのない主系のマシンとして、マスタ・ウィズアウト・スタンバイモードS_0で動作し、そのメモリ上に登録されたタスク・クラス５１，５２，…は、タスク・クラス編集機構９を用いて動的に追加・変更される。

【００４８】続いてマシンＢを立ち上げてイニシャルモードS_1にする。立ち上がったマシンＢよりマスタ・ウィズアウト・スタンバイモードS_0で動作しているマシンＡに対してタスク・クラス複製依頼を出し、その時点でのマシンＡのメモリ上に登録されたタスク・クラス５１，５２，…の複製を開始する。これでマシンＡはイコライズドモードS_eに、ＢはプリペアリングモードS_pに遷移する。これらのモードは過渡モードであり、タスク・クラス５１，５２，…の複製が終了すれば、マシンＡはマスタ・ウィズ・スタンバイモードS_mに、ＢはスタンバイモードS_bに自動的に遷移する。つまり、マシンＡはマシンＢによりバックアップされた状態の主系のマシンになり、マシンＢはマシンＡをバックアップする従系のマシンになる。

【００４９】このマシンＡがマスタ・ウィズ・スタンバイモードS_mに、ＢがスタンバイモードS_bになっている状態で、マシンＡにトラブルが発生し、マシンＡでのサービス続行が不可能になったとき、系切替えが発生する。この系切替えの発生によって、マシンＡは停止状態であるストップモードS_fに遷移し、マシンＢはマスタ・ウィズアウト・スタンバイモードS_0に遷移する。

【００５０】なお、マシンＡがマシンＢによりバックアップされた状態（マシンＡはマスタ・ウィズ・スタンバイモードS_m、マシンＢはスタンバイモードS_b）で、マシンＢにトラブルが発生した場合には、マシンＡはマスタ・ウィズアウト・スタンバイモードS_0に、マシンＢはストップモードS_fにそれぞれ遷移する。

【００５１】また、図４は２つのマシンのモード遷移を示す説明図である。この図４はマシンＡとマシンＢとが交互に主系となるモード遷移を、２つのマシンのモード遷移の点から示したものである。この場合、マシンＡがストップモードS_f、マシンＢがマスタ・ウィズアウト・スタンバイモードS_0の状態から、マシンＡがイニシャルモードS_1、マシンＢがマスタ・ウィズアウト・スタンバイモードS_0への矢印は、トラブルが発生したマシンＡの再立ち上げを行う場合のモード遷移を示しており、両方のマシンともストップモードS_fへの矢印は、マシンＡとマシンＢの双方にトラブルが発生した場合のモード遷移を示している。同様に、マシンＢがストップモードS_f、マシンＡがマスタ・ウィズアウト・スタンバイモードS_0の状態から、マシンＢがイニシャルモードS_1、マシンＡがマスタ・ウィズアウト・スタンバイモードS_0への矢印は、トラブルが発生したマシンＢの再立ち上げを行う場合のモード遷移を、両マシンともストップモードS_fへの矢印は、マシンＡとマシンＢの双方にトラブルが発生した場合のモード遷移を示している。

【００５２】また、マシンＡおよびマシンＢはそれぞれ時計回路１１を備えており、この時計回路１１はマシンＡとマシンＢの間で同期している。なお、この２つのマシン間で時計回路１１を同期化させるための技術は、例えば、Ｌ．Ｌａｍｐｏｒｔによる“Ｔｉｍｅ，Ｃｌｉｃｋｓ，ａｎｄ ｔｈｅ Ｏｒｄｅｒｉｎｇ ｏｆ Ｅｖｅｎｔｓ ｉｎ ａ Ｄｉｓｔｒｉｂｕｔｅｄ Ｓｙｓｔｅｍ”（Ｃｏｍｍｕｎｉｃａｔｉｏｎ ｏｆ ＡＣＭ，Ｖｏｌ．２１，Ｎｏ．７，ｐｐ．５５８－５６５，Ｊｕｌｙ，１９７８）などにおいて詳細に論じられている周知のものであるため、ここではその説明は割愛する。

【００５３】このように、時計回路１１をマシンＡとマシンＢとの間で同期させることにより、スケジュールが時刻ｂから周期ｃごとに生成されてきたとすると、ｎを整数として、時刻（ｂ＋ｎ・ｃ）に主系のマシンと従系のマシン上でスケジュールされる。スケジュール開始時刻に関する情報として、この時刻ｂと周期ｃが主系マシンから従系マシンに通知されれば、タスク・クラスの複製が時刻t_fに終了したとして、次式を満たす時刻t_sが、マシンＡとマシンＢの双方で同じスケジュールの作成を開始する時刻となる。また、実行されるタスクも全く同じ時刻に実行されることになる。

$$t_s = b + n_s \cdot c > t_f$$

【００５４】このように、この実施の形態１によれば、タスク・クラスが動的に変更されるような二重化システムにおいても、主系のマシンのタスク・クラスが複製された従系のマシンが、主系マシンのトラブルに対してホットバックアップ状態で準備しているので、主系マシンの異常時には従系マシンが即座に主系マシンに入れ替わってサービスを継続することが可能となり、これにより、２つのマシンを交互に主系とし、サービスを止めることなく、一方のマシンを保守することができる二重化システムが得られるとともに、マシン間で同期化された時計回路によって主系マシンと従系マシンの時刻を一致させているので、２つのマシンでサービスを切れ目なく

継続することが可能になるなどの効果がある。

【００５５】

【発明の効果】以上のように、この発明によれば、通信路で接続された２つのマシンのそれぞれに、動的に修正可能にタスク・クラスを保持させるとともに、モード遷移可能なアクティベータにおけるタスク・クラスの複製を行うモードで、タスク・クラスの複製を行うタスク・クラス複製機構と、実行時にタスク・クラスの編集を行うタスク・クラス編集機構とを設けるように構成したので、主系のマシンが正常な間にタスク・クラスを従系のマシンに複製して、主系マシンのトラブルに対して従系マシンがホットバックアップ状態で準備しておくことができて、タスク・クラスが動的に変更される二重化システムにおいても、主系のマシンの異常時には従系のマシンが即座に主系マシンに入れ替わってサービスを継続することが可能となり、また、これにより、２つのマシンを交互に主系とし、サービスを止めることなく、一方のマシンを保守することができる二重化システムが得られる効果がある。

【００５６】この発明によれば、２つのマシンのそれぞれに時計回路を持たせ、その時計回路をマシン間で同期させるように構成したので、主系マシンと従系マシンの時刻が一致し、２つのマシンでサービスを切れ目なく継続することができるという効果がある。

【図面の簡単な説明】

【図１】　この発明の実施の形態１による二重化システムを示す構成図である。

【図２】　実施の形態１における主系マシンと従系マシンの動作の概要を示す説明図である。

【図３】　実施の形態１における１つのマシンのモード遷移を示す説明図である。

【図４】　実施の形態１における２つのマシンのモード遷移を示す説明図である。

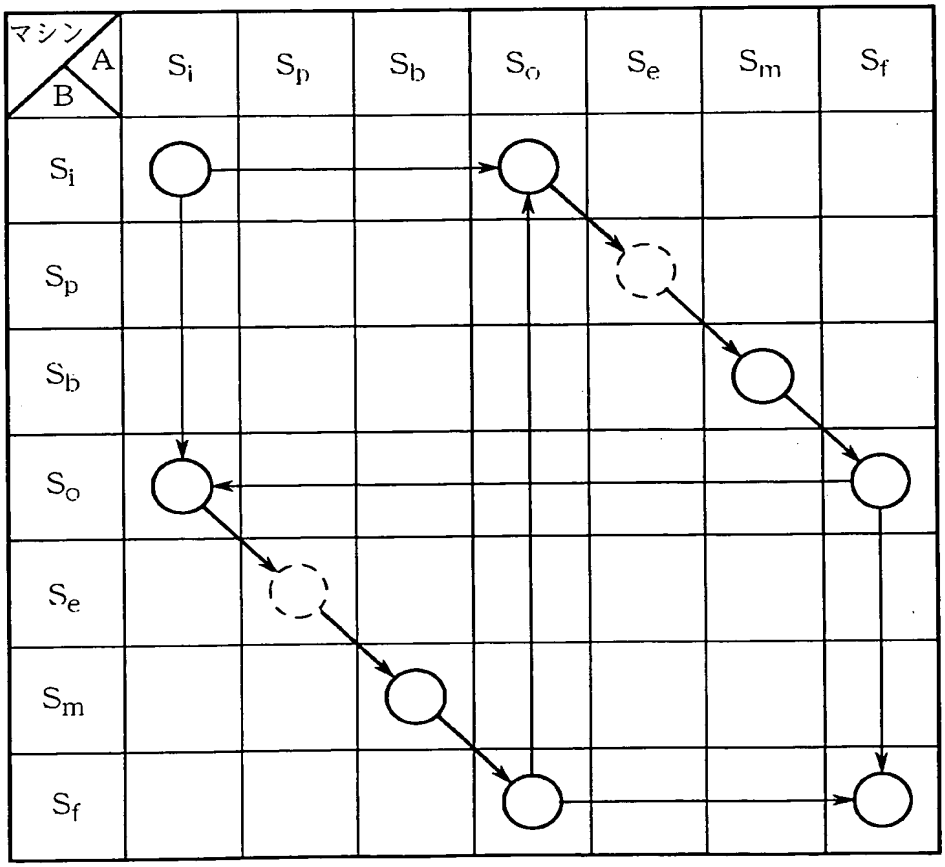
【図５】　従来の二重化システムを示す構成図である。

【図６】　従来の二重化システムにおけるアクティベータを示す説明図である。

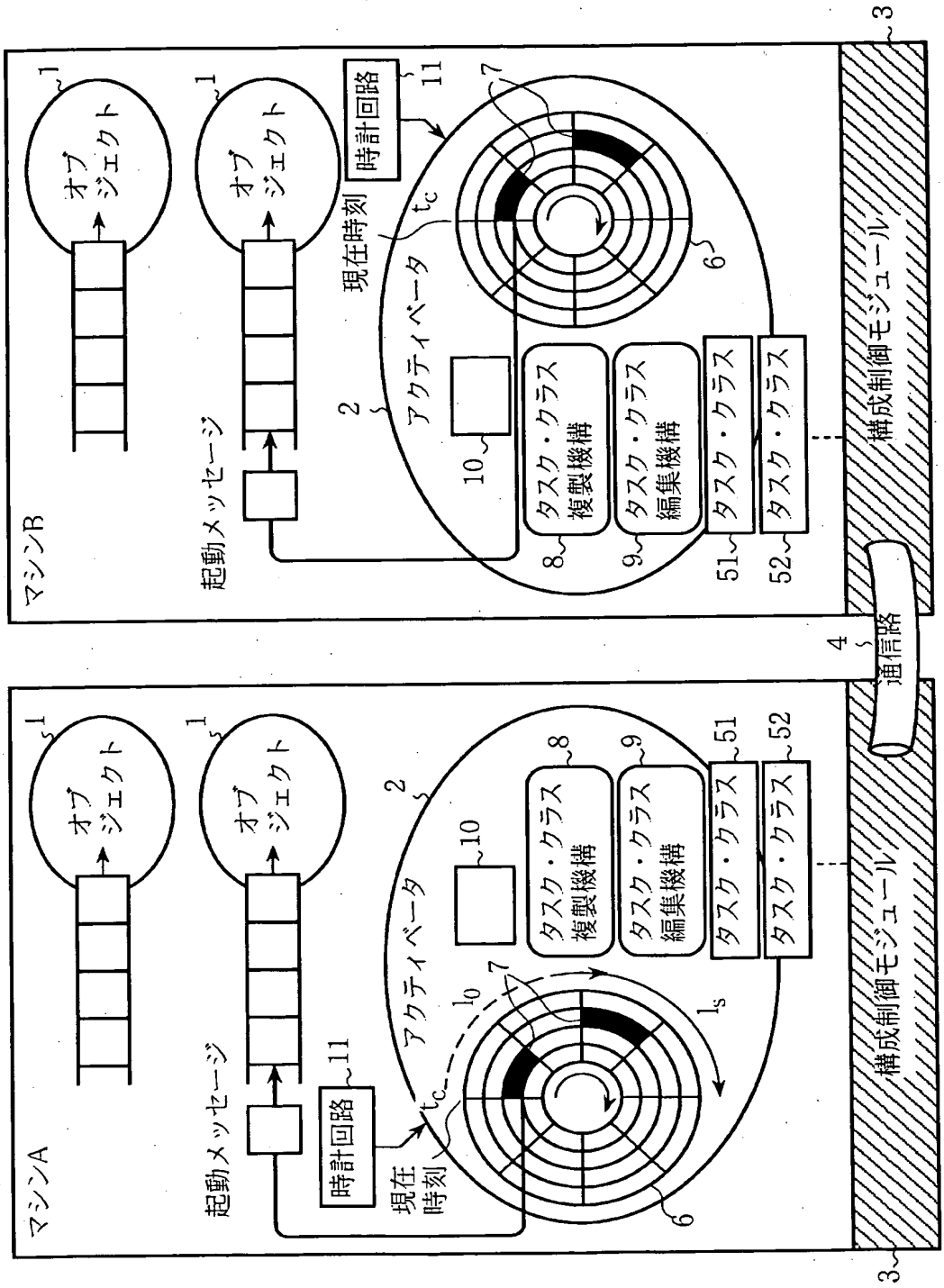
【符号の説明】

１　オブジェクト、２　アクティベータ、３　構成制御モジュール、４　通信路、５１，５２，…　タスク・クラス、８　タスク・クラス複製機構、９　タスク・クラス編集機構、１１　時計回路。

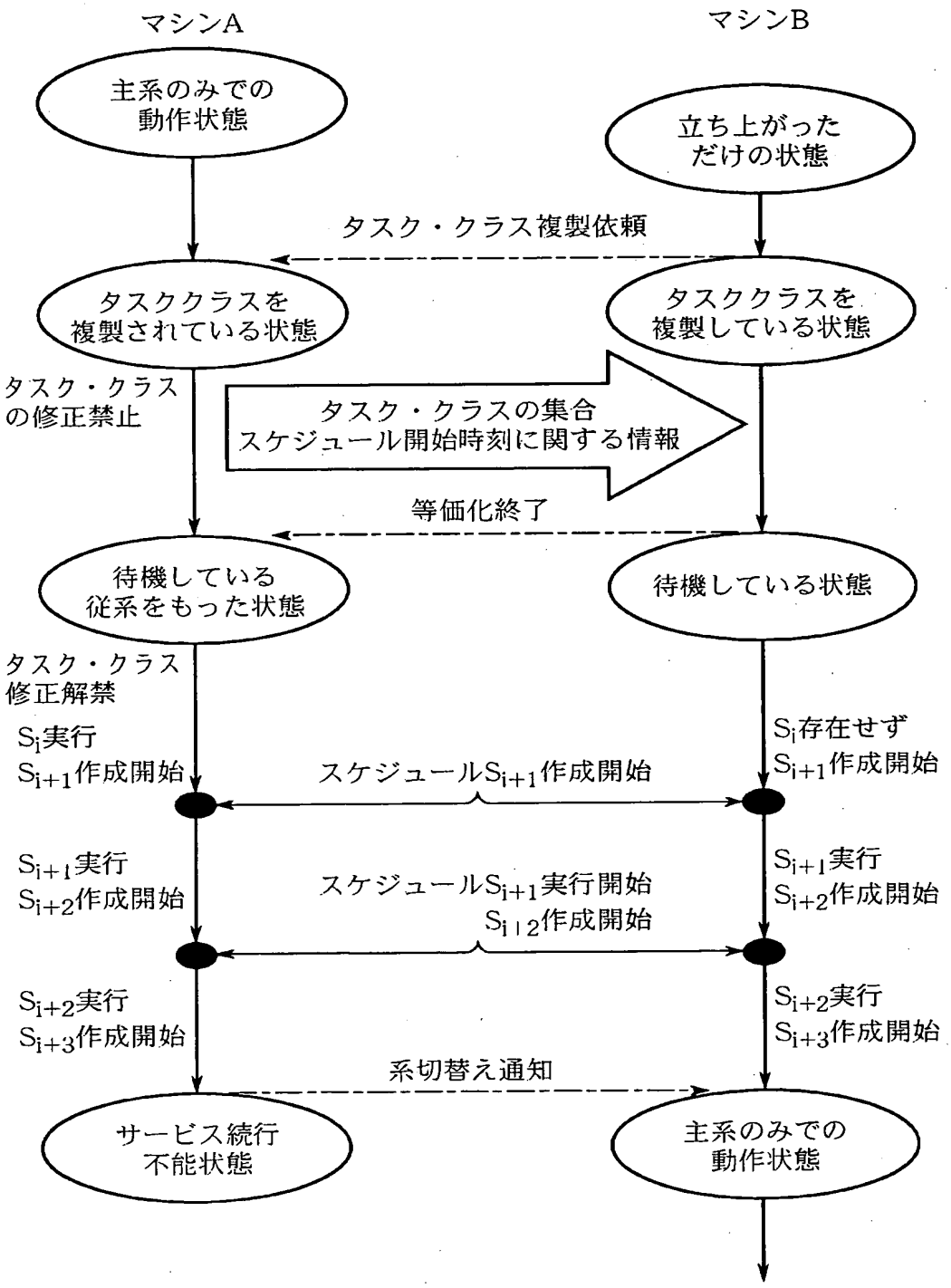
【図４】



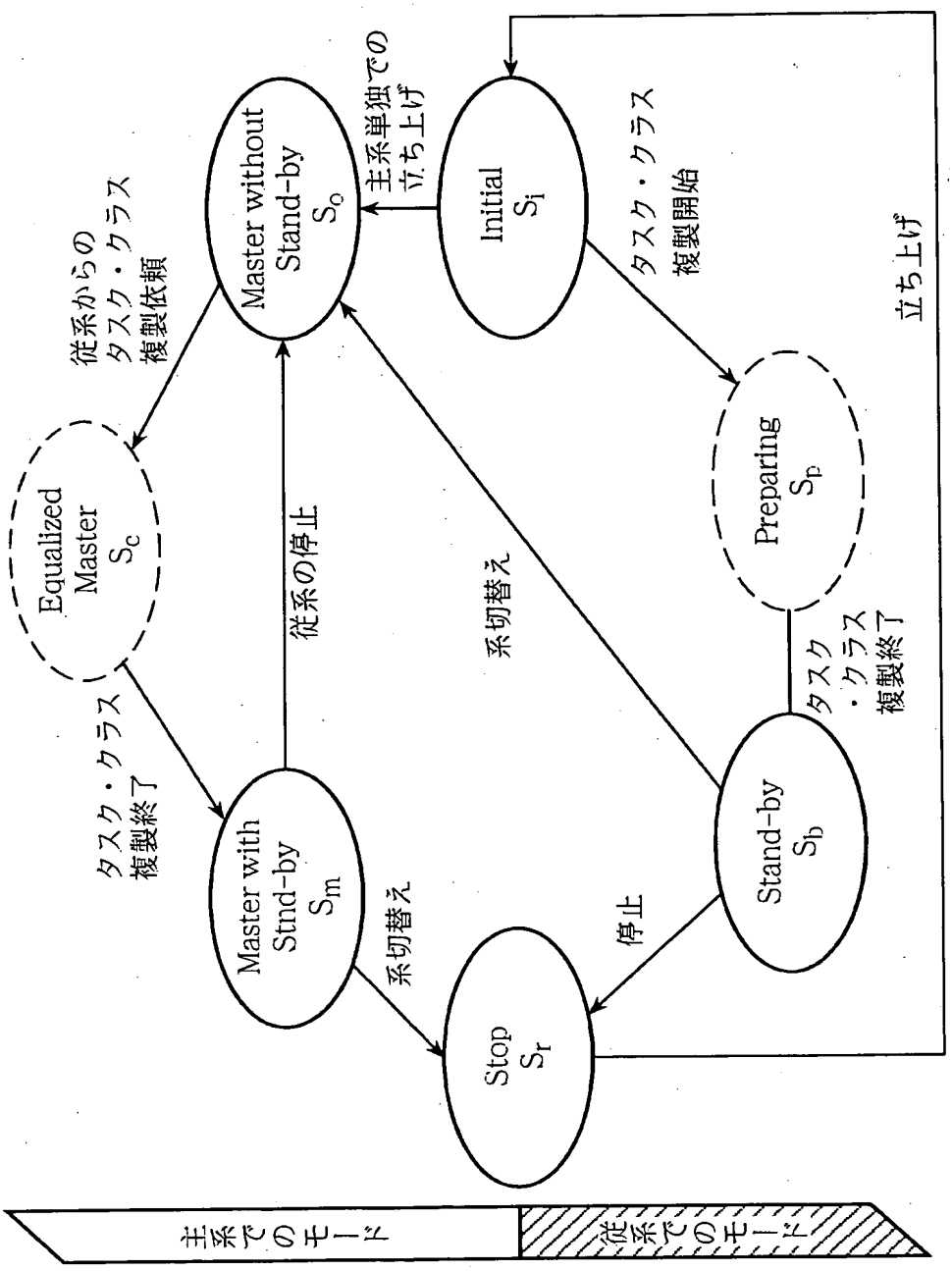
【図１】


マシンA
マシンB
オブジェクト
起動メッセージ
時計回路
現在時刻
アクティベータ
タスク・クラス
複製機構
タスク・クラス
編集機構
タスク・クラス
タスク・クラス
構成制御モジュール
通信路

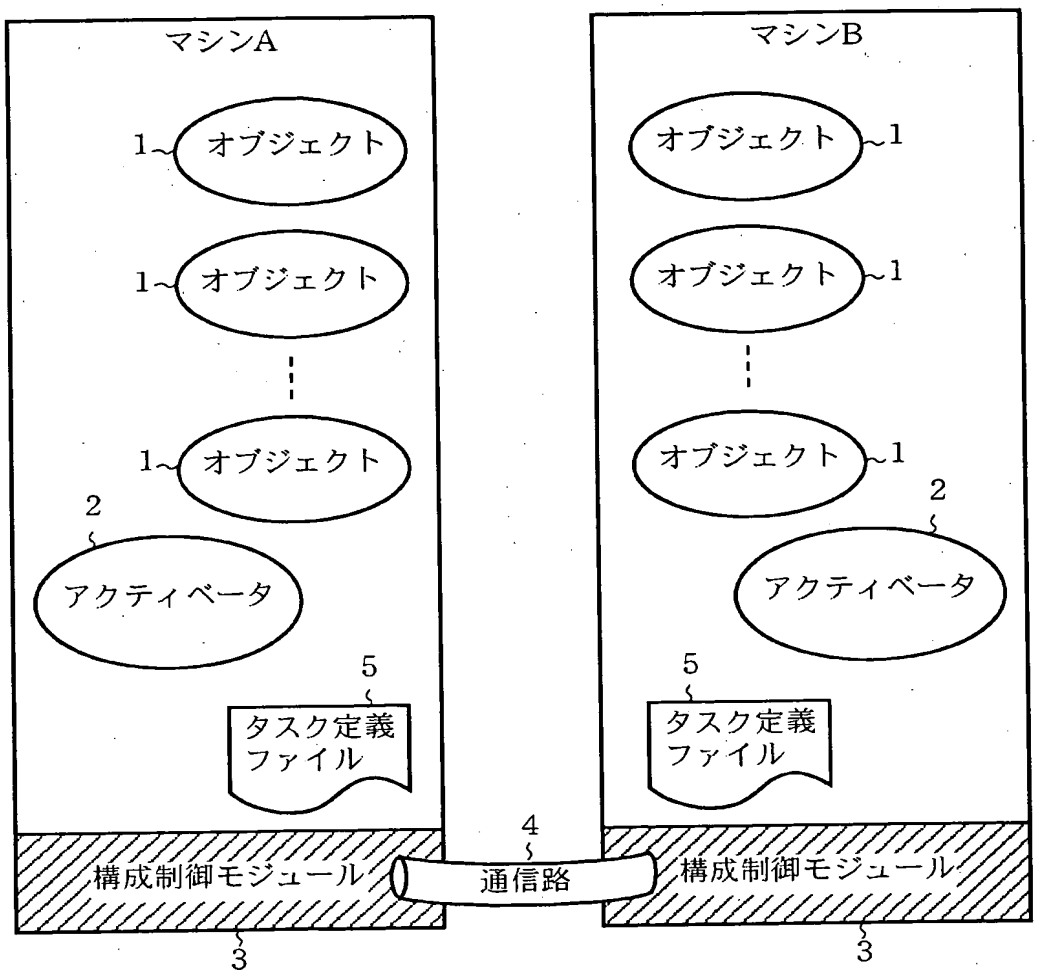
【図２】


マシンA
マシンB
主系のみでの
動作状態
立ち上がった
だけの状態
タスク・クラス複製依頼
タスククラスを
複製されている状態
タスククラスを
複製している状態
タスク・クラス
の修正禁止
タスク・クラスの集合
スケジュール開始時刻に関する情報
等価化終了
待機している
従系をもった状態
待機している状態
タスク・クラス
修正解禁
Si実行
Si+1作成開始
スケジュールSi+1作成開始
Si存在せず
Si+1作成開始
Si+1実行
Si+2作成開始
スケジュールSi+1実行開始
Si+2作成開始
Si+1実行
Si+2作成開始
Si+2実行
Si+3作成開始
Si+2実行
Si+3作成開始
系切替え通知
サービス続行
不能状態
主系のみでの
動作状態

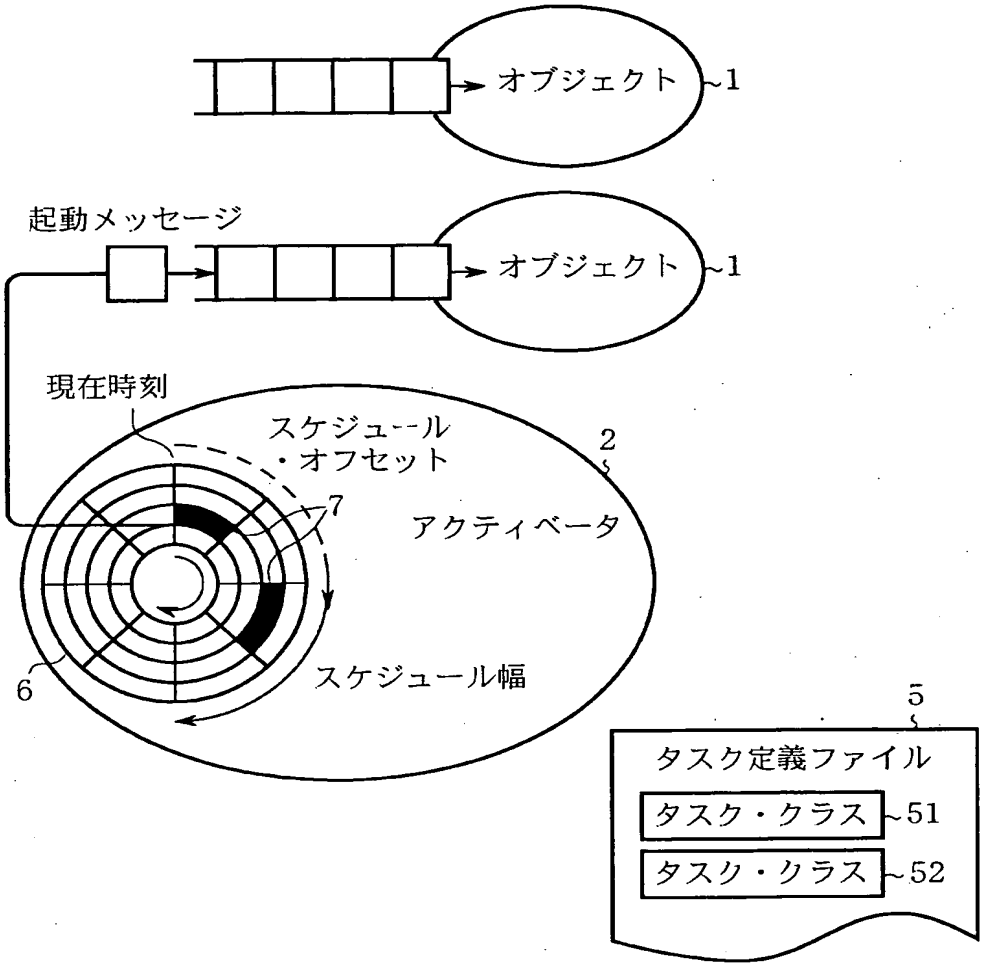
【図３】


Master without Stand-by S_o
主系単独での立ち上げ
Initial S_i
タスク・クラス複製開始
立ち上げ
従系からのタスク・クラス複製依頼
Equalized Master S_c
Preparing S_p
タスク・クラス複製終了
従系の停止
系切替え
タスク・クラス複製終了
Master with Stnd-by S_m
Stand-by S_b
系切替え
停止
Stop S_r
主系でのモード
従系でのモード

【図５】



【図６】


オブジェクト
1
起動メッセージ
オブジェクト
1
現在時刻
スケジュール
・オフセット
2
7
アクティベータ
6
スケジュール幅
5
タスク定義ファイル
タスク・クラス
51
タスク・クラス
52

CLAIMS

[Claim(s)]
[Claim 1] In the duplex system with which the machine of a ** system succeeds the service using two machines by which one side serves as a main system and another side serves as a ** system at the time of the abnormalities of the machine of a main system One or more objects in which mode transition is [performing a task] possible, The task class which was held dynamically possible [correction] on memory and which shows the set of said task which should be performed, Based on the task schedule created from said task class, tell the timing of task activation at said object, and mode transition is possible. The activator which has the mode which reproduces a task class between machines as one of the mode of the, The task class duplicator style which operates in the mode which reproduces said activator and reproduces the task class between machines, By the communication link with the task class edit device in which said task class is edited at the time of activation, and abnormality detection, by one's machine or a partner machine Duplex system characterized by connecting by the channel for giving each of said machine the configuration control module which determines the mode in which its machine should change, and notifying a mutual working state for these two machines of each other's.
[Claim 2] Duplex system according to claim 1 characterized by each machine having the clock circuit which synchronized between two machines.

[Translation done.]
* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the duplex system with which the trouble in a system enables it to use independently the function of data processing system to perform a task at the reserved time of day, continuously from the viewpoint of a user periodically.
[0002]
[Description of the Prior Art] In machines, such as a calculating machine which constitutes data processing system, abnormalities are in a disk, and suppose that the data writing to a disk went wrong while being count. In this case, by the machine concerned, continuation of an activity may become difficult more than it. In such a case, the system is doubled by Machine

A and Machine B, for example, if continuation of an activity becomes difficult more than it by Machine A, continuation of an activity will be requested to Machine B. Even if a trouble is in the machine which is one side even if in this way, it must enable it to use one system continuously from the viewpoint of a user in the system by which high dependability is required of a processing result, as an activity with the succeedingly same machine of another side is done.

[0003] In the duplex system which consists of two machines, such a machine A and Machine B, the machine of a main system and another side serves as [one machine] a ** system, only the machine of a main system usually operates, and the machine of a ** system is in a standby condition. This is because conflict of duplication of an activity etc. must not arise when a ** system machine works while the main system machine is working. When a trouble occurs in the machine used as a main system and continuation of an activity becomes impossible in it more than it, the machine which had become a ** system till now serves as a main system, and continues activity activation.

[0004] The conventional duplex system is divided roughly into two, the cold standby method which starts the machine of another side and succeeds service when one machine lapses into the abnormality situation, and the hot standby method with which the machine of another side succeeds the service immediately when the task with two always same machines is performed and one machine lapses into the abnormality situation.

[0005] Drawing 5 is the block diagram showing the duplex system by the conventional cold standby method shown in the "1 design technique of the stand-by redundant system which guarantees real-time constraint" (Institute of Electronics, Information and Communication Engineers paper magazine, Vol.J78-D-I, No.8, 1995.8) by Ichiro Mizunuma, **** Hiroo, Hiromitsu Shimakawa, and the bamboo rail fence peak 1. For an object and 2, as for a configuration control module and 4, in drawing, an activator and 3 are [1 / a channel and 5] task definition files.

[0006] Moreover, drawing 6 is the explanatory view showing the configuration of the activator 2, gives the same sign as drawing 5 to a considerable part, and omits the explanation. In drawing, 6 is a schedule table in the activator 2 concerned, 7 is a task instance in the schedule table 6, and 51, 52, and -- are the task classes in the task definition file 5.

[0007] Next, actuation is explained. First, the service taking over in the duplex system of the conventional cold standby method is explained. In addition, use Machine A as the machine of a main system in this case, and let Machine B be the machine of a ** system. If the configuration control module 3 of the machine B of a ** system receives the notice of taking over from the machine A of a main system or it becomes impossible to receive the survival message from the machine A of a main system now, it will judge that activity continuation by the machine A of a main system is impossible, and service provision will be started using the object 1 on its machine (machine B). Initiation of this service taking over reads the contents of the task definition file 5 on Machine B, and is realized by starting the activator 2 which creates a schedule periodically.

[0008] Next, service taking over with the duplex system of the conventional hot standby method is explained. When the hot standby method makes the machine of a ** system always perform the completely same actuation as the machine of a main system and abnormalities arise in the machine of a main system, the machine of a ** system enables it to succeed the service immediately. By this hot standby method, a main system machine and a ** system machine are started by coincidence, and those objects 1 and an activator 2 take the completely same actuation. Moreover, the task classes 51 and 52 in the task definition file 5 and correction of -- are also made on coincidence by the main system machine and the ** system machine, respectively.

[0009] In the duplex system of such a hot standby method, although the machine of a main

system and the machine of a ** system are working to coincidence, when these [both] send out output data, the same data will overlap, the candidate for an output will be reached, and derangement may be caused. Therefore, even if an object 1 can have the mode and the machine of a ** system receives the same message, although the same count as the machine of a main system is performed, it is made not to perform the output of data.
[0010] Moreover, in the duplex system of a cold standby method, in order for a ** system machine to newly start actuation as a main system machine at the time of taking over of service, it is necessary to deal with read-out of a file, initialization of memory, etc. in many cases. For this reason, in case an object 1 changes the mode, it is common to enable it to perform procedure specified beforehand.
[0011]
[Problem(s) to be Solved by the Invention] In case the machine of a ** system newly tended to turn into a machine of a main system, the task classes 51 and 52 and -- were read from the task definition file 5, and since the duplex system of the conventional cold standby method was constituted as mentioned above, when corrected by the task classes 51 and 52 and the machine whose -- was a main system till then at the time of activation, the technical problem that it could not be coped with occurred.
[0012] In addition, in this way, in case a ** system machine newly tends to turn into a main system machine As long as the task classes 51 and 52 and the method which reads -- are adopted from the task definition file 5, in order [of the task classes 51 and 52 and --] to correspond to dynamic correction The task definition file 5 in a main system machine and a ** system machine by a help's etc. approach Even if it is required to maintain so that the contents may always be in agreement and it performs processing for making these contents in agreement, in the duplex system of the conventional cold standby method Since an activator 2 was started by the ** system machine after the abnormalities of a main system machine are discovered, the technical problem that it took time amount by service initiation by the ** system machine from the abnormal occurrence in a main system machine also occurred.
[0013] On the other hand with the duplex system of the conventional hot standby method In a system by which a task is dynamically changed and service without resting is required since the machine of a main system and the machine of a ** system must always take the completely same actuation Since abnormalities arose in the main system machine, when another abnormalities arose in the machine which newly became a main system, the technical problem that it could not be coped with since there is no machine which has backed this up occurred.
[0014] It was made in order to solve the above technical problems, and a task class is held on memory, and this invention can correct this dynamically. By copying a task class in the machine of the ** system of a cold standby condition from the main system machine of an all seems well The machine of a ** system is changed into the hot standby condition, and when abnormalities occur in the machine of a main system, it aims at obtaining the duplex system which enabled it to exchange the machine of a ** system immediately.
[0015]
[Means for Solving the Problem] The duplex system concerning this invention connects two machines by the channel, and sets them in each [these] machine. The task class which shows the set of the task which should be performed is held dynamically possible [correction] on memory. Based on the task schedule created from the task class, mode transition of the activator which tells an object about the activation timing of a task is enabled. As one of the mode of the In the mode which reproduces the task class in an activator to each of two machines while giving the mode which reproduces a task class The task class duplicator style which reproduces a task class, and the task class edit device in which a task class is edited at the time of activation are established.

[0016] The duplex system concerning this invention gives the clock circuit which synchronized between two machines to each machine.

[0017]

[Embodiment of the Invention] Hereafter, one gestalt of implementation of this invention is explained.

Gestalt 1. drawing 1 of operation is the block diagram showing the duplex system by the gestalt 1 of implementation of this invention. In drawing, 1 is at least one object which performs a task, and mode transition is possible for this object 1. 2 is an activator which tells this object 1 about the starting timing of task activation, and this activator 2 is the point of having the mode in which transition is possible and the mode reproduces a task class between machines as one of the mode of that, and differs from conventional it. 3 is a configuration control module which is always inspecting whether the treatment is managed to the notice of abnormalities by its machine, and a partner's machine is working normally. 4 is a channel for this configuration control module 3 to notify the operating status of a mutual machine mutually.

[0018] Moreover, 51, 52, and -- are task classes which show the set of the task performed by the object 1, are the point that what is registered into the task definition file which omitted illustration in this case is what is held dynamically possible [correction] on memory at the time of starting of a machine, and differ from conventional it. It is the schedule table in which 6 being in an activator 2 and showing the activation timing of a task, and 7 is a task instance registered on this schedule table 6. 8 operates in the mode which reproduces a task class between the machines which an activator 2 has, and is the task classes 51 and 52 between machines, and a task class duplicator style of -- which reproduces. 9 is a task class edit prohibition flag which are the task classes 51 and 52 and the task class edit device of -- in which it edits, and becomes effective [10] in case [of these task classes 51 and 52 and --] edit is forbidden at the time of activation. 11 is a clock circuit which generates the information on current time, and is synchronized between two machines.

[0019] This duplex system is constituted by two machines, the machine A identically formed of these, and Machine B.

[0020] Here, work of each component is explained. First, an object 1 is explained. The object 1 consists of methods which showed the internal memory and the count approach for holding an intermediate count condition, and when a start up message is received from the exterior, it starts the method specified in the start up message. The started method calculates using an internal memory. It confirms whether if activation of a method finishes, the start up message with a new object 1 has reached oneself, and if it has arrived, a method will be started similarly. It is dormant until the following start up message arrives, if it has not arrived. An object 1 may perform actuation which is different depending on the mode even if having the mode is allowed and it receives the same message. Moreover, in case the mode is changed, it can also be specified that it performs the specified method. In addition, about this, it mentions later.

[0021] Next, the task classes 51 and 52 and -- are explained. The task classes 51 and 52 and -- show the set of the task which should be performed, respectively. In addition, by the machine of a main system, and the machine of a ** system, these task classes 51 and 52 and the contents of -- are usually the same. Here, there are a periodic task and a starting time designated task as task. By the periodic task, a message, a starting period, etc. which should be sent to the identifier of the object 1 which should be performed, and an object 1 are specified, and a message, starting time of day, etc. which should be sent to the identifier of the object 1 which should be performed, and an object 1 are specified by the starting time designated task.

[0022] In addition, when there are more than one, the starting time of day of the

above-mentioned starting time designated task may be specified as a list of starting time of day, and may be specified as a set of starting time of day using a certain algebraic-expression expression. As an example of the latter which specifies the set of starting time of day using a certain algebraic-expression expression, a thing like for example, a degree type can be considered.
{c|c ** WORKINGDAY ** c=NOON}
This expresses the noon of the working day except a holiday, and it is shown that a task is started at this time of day.
[0023] Even if it makes it these periods task and makes it a starting time designated task, the set of the task actually started will be expressed. Then, two are distinguished from the task instance (each task actually started) 7 as the task classes (set of the task expressed by assignment of starting conditions) 51 and 52, and --.
[0024] Next, an activator 2 is explained. An activator 2 determines the task classes 51 and 52 and the task instance 7 of -- which should be periodically started in the near future according to the contents, transmits a start up message to each object 1, and starts the task instance 7. The activator 2 is equipped with the schedule table 6 in order to realize this. This schedule table 6 is tc about current time. Period Iw which is carried out and is shown by the degree type The schedule of the task instance 7 which should be started shall be held.
Iw :[tc ,tc +w)
[0025] An activator 2 is the period Is which reads the task classes 51 and 52 and -- periodically, and is shown by the degree type. The task instance 7 which should be carried out a schedule is determined, and it is written in the schedule table 6.
Is :[tc +lo ,tc +lo +ls)
[0026] Here, they are lo and ls. It will be called schedule offset and schedule width of face, respectively. In addition, schedule offset lo It is the difference of the starting point of the period of the near future and schedule activity start time which should be carried out a schedule, and is the schedule width of face ls. It is the die length of the period of the near future which should be carried out a schedule.
[0027] After an activator 2 ends a schedule, the task instance 7 by which warm-up time is set as current time investigates whether it is registered on the schedule table 6. If registered, the start up message specified as the object 1 specified by the task instance 7 will be transmitted, and a task will be started.
[0028] When abnormalities occur in the machine of a main system, the machine of a ** system can succeed the service immediately here. In order to enable it to succeed the ** system machine which suspended dynamically service of the task classes 51 and 52 and the main system machine by which -- is corrected always, and with the gestalt of this operation The function which reproduces the task classes 51 and 52 of a main system and -- in the machine of a ** system is given to the activator 2. That is, the activator 2 is equipped with the task class duplicator style 8, and reproduces the task classes 51 and 52 on a main system machine, and -- on a ** system machine using it in the activator 2 of a ** system machine. Moreover, it also has the task class edit device 9 and the task class edit prohibition flag 10, and the task classes 51 and 52, an addition, deletion, correction of --, etc. are dynamically edited by the task class edit device 9. In addition, he forbids edit of the task classes 51 and 52 by the above-mentioned task class duplicator style 8, and -- at the time of a duplicate, confirm the task class edit prohibition flag 10, and according to the task class edit device 9, and is trying for conflict not to produce it in the task classes 51 and 52 on the reproduced ** system machine, and --.
[0029] Next, the configuration control module 3 is explained. The configuration control module 3 manages management when there are abnormalities in the I/O unit of those with one, and each machine etc. in each machine. It is divided roughly into two of detection of the

abnormalities in a partner machine judged that work of this configuration control module 3 had abnormalities in the partner machine when nothing had been heard from the notice of the abnormalities in a self-machine which tell the abnormalities on their machine to the configuration control module 3 on a partner machine, and the partner machine. In addition, even if not unusual, when a user changes the machine of a main system to the machine of a ** system intentionally for maintenance etc., it considers that the system change signal from a user is generating of a false abnormal condition, and when special [in the abnormalities in a machine], it treats [******].

[0030] As an example in the case of the abnormalities in a self-machine, abnormalities are in the disk of Machine A, it supposes that the object 1 of one of the machines A failed in the data writing to a disk, and the case where continuation of an activity is difficult is considered more by Machine A. In this case, taking over of service in Machine B must be requested. The object 1 which failed in the data writing to a disk on Machine A notifies that to the configuration control module 3 of its machine. The configuration control module 3 which received the notice notifies using a channel 4 that oneself (machine A) cannot continue an activity any more to the configuration control module 3 of the partner machine B, i.e., a machine. This notice will be called "a notice of taking over." The configuration control module 3 on the machine B which received the notice of taking over starts performing a task using the object 1 on Machine B.

[0031] Next, abnormality detection and its management of a partner machine are explained. Abnormality detection of a partner machine is detected by having heard nothing. In order to detect both abnormalities, the configuration control module 3 has sent out the "survival message" to the configuration control module 3 of a partner machine periodically through the channel 4. As long as this survival message has arrived periodically, it can be judged that the partner machine is operating normally. However, when a survival message stops arriving from a partner machine, it is thought that the machine which became having not heard from [nothing] has lapsed into the abnormal condition to the extent that having become impossible of operation is not even told. When a survival message stops arriving from a main system machine, as for a ** system machine, a main system machine starts performing a task using the object 1 on one's machine, without judging that actuation is already impossible and waiting for the notice of taking over.

[0032] Moreover, a channel 4 is used in order that two machines may notify or detect mutual abnormalities as mentioned above, and it is realized by a hardwired line, the network, etc.

[0033] Next, actuation is explained. Here, the outline of actuation of a main system machine and a ** system machine is shown to drawing 2. In the example of the illustration to this drawing 2 R> 2, Machine A shall be [Machine B] a ** system by the main system.

[0034] First, the machine B of a ** system should once stop according to factors, such as maintenance. In the meantime, the machine A of a main system does not have the ** system machine which is standing by, and is in "the operating state only in a main system." Then, the machine B of a ** system starts. At this time, it does not have the task classes 51 and 52 and -- which showed whether the schedule of what kind of task should have been carried out in the machine B of a ** system. In order to be able to perform service taking over of the machine A of a main system, the machine B which is a ** system requests the duplicate of the task classes 51 and 52 and -- from Machine A through a channel 4, and shifts to "the condition of having reproduced the task class." The machine A of the main system which received this request shifts to "the condition that a task class is reproduced."

[0035] If the machine A of a main system will be in "the condition that a task class is reproduced" and the machine B of a ** system will be in "the condition of having reproduced the task class", the task classes 51 and 52 of Machine A and the set of -- will be reproduced by the machine B of a ** system. Reproduction of these task classes 51 and 52 and -- is

performed using the task class duplicator style 8 on each machine. Together with these task classes 51 and 52 and --, the information about schedule start time is also then notified to the machine B of a ** system from the machine A of a main system. Supposing the schedule has been generated by every period c from time of day b, the schedule of it will be carried out to time of day (b+n-c) on the machine A of a main system by making n into an integer. As information about this schedule start time, if these time of day b and a period c are notified to Machine B from Machine A, the duplicate of a task class will be time of day tf. Time of day ts which fills a degree type noting that it ends The time of day which starts creation of the same schedule on the both sides of Machine A and Machine B comes.
ts =b+ns and c>tf [0036] in this case -- even if the clock circuit 11 does not synchronize between the clock circuit 11 of the machine A of a main system, and the machine B of a ** system -- ready -- several ns By choosing a value appropriately on each machine, it is the worst and scheduling is started to the timing from which only the period c shifted.
[0037] In addition, in the "condition that a task class is reproduced" of the machine A of this main system, and the "condition of having reproduced the task class" of the machine B of a ** system, edit of the task classes 51 and 52 and -- is forbidden by confirming the task class edit prohibition flag 10. If the task classes 51 and 52 and -- have an addition, deletion, correction, etc. then, since conflict will arise in the reproduced task class, this has forbidden edit of the task classes 51 and 52 and --, in order to prevent it.
[0038] After the duplicate of the task classes 51 and 52 by the task class duplicator style 8 and -- is completed, the equivalence of the task classes 51 and 52 and the condition of -- is carried out by Machine A and Machine B, and Machine B uses equivalence termination and tells Machine A about a channel 4. Thereby, "it will have the ** system which is standing by" the machine A of a main system, and the machine B of a ** system will be in "the condition of standing by." In these condition, Machine A and Machine B perform the computation same as [both] internal count. In order to make it there be no output which overlapped to the exterior then, actuation of only the output of a count result being controlled is taken by the machine B of a ** system.
[0039] In the "condition of having had the ** system which is standing by" of the machine A of a main system, and the "condition of standing by" of the machine B of a ** system, if the schedule of the task of the period in the near future is carried out and the period visits, a task will be started according to a schedule. For example, the task of a previous period is started according to the schedule from which it was started at time of day (b+n-c), and creation of the schedule of the task performed by period [b+n-c and b+(n+1) -c was made here.
[0040] Now, a trouble occurs by the machine A of a main system, and suppose that continuation of service by Machine A became impossible. This is detected with the configuration control module 3 of either the machine A of a main system, or the machine B of a ** system. When detected by the configuration control module 3 of the machine A of a main system, the notice of a system change is sent to the machine B of a ** system from the machine A of a main system. Moreover, when detected by the configuration control module 3 of the machine B of a ** system, Machine B issues the notice of a system change to oneself. Since the machine B which received the notice of a system change has the same schedule as Machine A, service of Machine A can be succeeded immediately. Although Machine B turns into a main system machine at this time and it operates, that condition is "operating state only in a main system."
[0041] It is constituted as a list [of the task classes 51 and 52 by which current registration is carried out on memory, and --] of the element with which a set expresses the task classes 51 and 52 and --. The task class edit device 9 corrects task classes, such as deletion of a task class [finishing / the addition of a new task class and registration / to a set] of the current task classes 51 and 52 and --, dynamically to this list at the time of activation. For example, the

addition of the new task class to the list representation of a task class can be realized by element addition on a list, and the retrieval and deletion of an element from a list can realize deletion of a task class [finishing / registration]. However, that edit is allowed [of the task classes 51 and 52 and --] is only the case that the task class edit prohibition flag 10 is invalid. When the task class edit prohibition flag 10 is effective, edit of the task classes 51 and 52 and -- goes wrong, or is postponed.
[0042] The task class duplicator style 8 has each method used by the main system and ** system side. The task class duplicator style 8 confirms the task class edit prohibition flag 10 first, according to whether subsequently oneself is a main system or it is a ** system, performs each method by the side of a main system or a ** system, and makes an invalid the task class edit prohibition flag 10 after that. In the method by the side of a main system, the task classes 51 and 52 and -- which are registered are notified to a ** system side, and the sent task class is registered by the method by the side of a ** system. The method of the task class duplicator style 8 by the side of a main system reads the lists of elements showing the task classes 51 and 52 and -- one by one, and is realized by transmitting this to a ** system side using a channel 4. Moreover, the methods of the task class duplicator style 8 by the side of a ** system are realized one by one in the sent task class by [of the task classes 51 and 52 and --] adding to list representation and going, for example.
[0043] Here, this activator 2 changes the mode [some (in this case, seven sorts)], in order to reproduce the task classes 51 and 52 and --. The modes are enumerated below. In addition, in order to simplify explanation, make Machine A into a main system also here, and let Machine B be a ** system.
[0044] Sf : stop (Stop) mode Si in which the machine has not started : A machine starts. The task classes 51 and 52 and -- are the initial (Initial) mode Sp which is not registered yet. : It is in transient mode in which the machine B of a ** system has copied the task class of the machine A of a main system. PURIPEA ring (Preparing) mode Sb : On the machine B of a ** system, in the task classes 51 and 52 and the mode of -- which the duplicate ended Standby (Standby) mode So of which oneself can relieve the machine A of a main system at any time : Carry out actuation only by the main system. a master and the Wiz -- out standby (Master-without-standby) mode Se : the machine A of a main system is in transient mode in which a task class is copied by the machine B of a ** system -- IKORAIZUDO master (Equalized-master) mode Sm : In the mode of the machine of a main system the master and the Wiz which oneself is working and the machine of a ** system has backed up - standby (Master-with-standby) mode [0045] In addition, an activator 2 is the PURIPEA ring mode Sp which is transient mode. Or IKORAIZUDO master mode Se If the duplicate of the task classes 51 and 52 and -- is completed, it will change automatically in the following mode.
[0046] Next, mode transition of this activator 2 is explained to a detail. Here, drawing 3 is the explanatory view showing mode transition of one machine. First, Machine A is made into a main system, and by making Machine B into a ** system, operation initiation is carried out and it illustrates how duplex system acts.
[0047] In order to work duplex system, Machine A and Machine B are started first. the mode at this time -- initial mode Si it is . subsequently, the master of a main system simple substance without backup of Machine A and the Wiz -- out standby mode So It is made to change. Here, the task classes 51 and 52 and -- which are held at the task definition file which omitted illustration, for example are registered dynamically possible [edit] on memory. henceforth -- as the machine of the main system in which Machine A does not have the machine which backs up -- a master and the Wiz -- out standby mode So It operates, and the task class edit device 9 is used for the task classes 51 and 52 and -- which were registered on the memory, and they are added and changed dynamically.
[0048] Then, Machine B is started and it is the initial mode Si. It carries out. the machine B

which started -- a master and the Wiz -- out standby mode So A task class duplicate request is issued to the machine A which is operating, and the reproduction of the task classes 51 and 52 registered on the memory of the machine A in the time and -- is started. Machine A is equalizing DOMODO Se now. B is the PURIPEA ring mode Sp. It changes. if these modes are transient modes and the duplicate of the task classes 51 and 52 and -- is completed -- Machine A -- a master and the Wiz - standby mode Sm B -- standby mode Sb It changes automatically. That is, Machine A turns into a machine of the main system in the condition of having been backed up by Machine B, and Machine B turns into a machine of the ** system which backs up Machine A.
[0049] this machine A -- a master and the Wiz - standby mode Sm B -- standby mode Sb When a trouble occurs in Machine A and service continuation by Machine A becomes impossible in the condition of having become, a system change occurs. stop mode Sf whose machine A is a idle state by generating of this system change changing -- Machine B -- a master and the Wiz -- out standby mode So It changes.
[0050] in addition -- the case where are in the condition (machine A a master, the Wiz the - standby mode Sm and Machine B standby mode Sb) that Machine A was backed up by Machine B, and a trouble occurs in Machine B -- Machine A -- a master and the Wiz -- out standby mode So Machine B -- stop mode Sf It changes, respectively.
[0051] Moreover, drawing 4 is the explanatory view showing mode transition of two machines. This drawing 4 is shown from the point of the mode transition of two machines by the mode transition to which Machine A and Machine B serve as a main system by turns. in this case, the machine A -- a stop mode Sf and Machine B -- a master and the Wiz -- out standby mode So From a condition Machine A -- the initial mode Si and Machine B -- a master and the Wiz -- out standby mode So An arrow head The mode transition in the case of performing re-starting of the machine A which the trouble generated is shown, and both machines are stop modes Sf. The arrow head shows the both sides of Machine A and Machine B mode transition when a trouble occurs. the same -- Machine B -- a stop mode Sf and Machine A -- a master and the Wiz -- out standby mode So From a condition Machine B -- the initial mode Si and Machine A -- a master and the Wiz -- out standby mode So An arrow head About the mode transition in the case of performing re-starting of the machine B which the trouble generated, both machines are stop modes Sf. The arrow head shows the both sides of Machine A and Machine B mode transition when a trouble occurs.
[0052] Moreover, Machine A and Machine B are equipped with the clock circuit 11, respectively, and this clock circuit 11 synchronizes between Machine A and Machine B. In addition, the technique for synchronizing the clock circuit 11 between these two machines For example L. "Time by Lamport, Clicks, and and the Ordering of Events in a Distributed In System" (Communication of ACM, Vol.21, No.7, pp.558-565, July, 1978) etc. Since it is the thing of the common knowledge currently discussed by the detail, the explanation is omitted here.
[0053] Thus, supposing the schedule has been generated by every period c from time of day b by synchronizing the clock circuit 11 between Machine A and Machine B, a schedule will be carried out to time of day (b+n-c) on the machine of a main system, and the machine of a ** system by making n into an integer. As information about schedule start time, if this time of day b and period c are notified to a ** system machine from a main system machine, the duplicate of a task class will be time of day tf. Time of day ts which fills a degree type noting that it ends The time of day which starts creation of the same schedule on the both sides of Machine A and Machine B comes. Moreover, the task performed will also be performed at the completely same time of day.
ts =b+ns and c>tf [0054] Thus, according to the gestalt 1 of this operation, it also sets to duplex system with which a task class is changed dynamically. Since the machine of a **

system by which the task class of the machine of a main system was reproduced is preparing in the state of hot backup to the trouble of a main system machine It becomes possible for a ** system machine to interchange in a main system machine immediately at the time of the abnormalities of a main system machine, and to continue service. By this While the duplex system which can maintain one machine is obtained without making two machines into a main system by turns, and stopping service Since the time of day of a main system machine and a ** system machine is made in agreement by the clock circuit synchronized between machines, there is effectiveness, like it becomes possible to continue service without a break by two machines.
[0055]
[Effect of the Invention] As mentioned above, while making a task class hold dynamically possible [correction] to each of two machines connected by the channel according to this invention In the mode which reproduces the task class in the activator in which mode transition is possible Since it constituted so that the task class duplicator style which reproduces a task class, and the task class edit device in which a task class is edited at the time of activation might be established While the machine of a main system is normal, a task class is reproduced in the machine of a ** system. Also in the duplex system with which a ** system machine can prepare in the state of hot backup to the trouble of a main system machine, and a task class is changed dynamically It becomes possible for the machine of a ** system to interchange in a main system machine immediately at the time of the abnormalities of the machine of a main system, and to continue service. By this It is effective in the duplex system which can maintain one machine being obtained, without making two machines into a main system by turns, and stopping service.
[0056] since according to this invention it constituted so that two machines might be alike, respectively, a clock circuit might be given and that clock circuit might be synchronized between machines, the time of day of a main system machine and a ** system machine is in agreement, and it is effective in service being continuable without a break by two machines.

[Translation done.]
* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram showing the duplex system by the gestalt 1 of implementation of this invention.
[Drawing 2] It is the explanatory view showing the outline of actuation of the main system machine in the gestalt 1 of operation, and a ** system machine.
[Drawing 3] It is the explanatory view showing mode transition of one machine in the gestalt 1 of operation.
[Drawing 4] It is the explanatory view showing mode transition of two machines in the gestalt 1 of operation.
[Drawing 5] It is the block diagram showing the conventional duplex system.
[Drawing 6] It is the explanatory view showing the activator in the conventional duplex

system.
[Description of Notations]
1 An object, 2 An activator, 3 A configuration control module, 4 A channel, 51 and 52, -- A task class, 8 A task class duplicator style, 9 A task class edit device, 11 Clock circuit.

[Translation done.]
* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

[Drawing 4]

[Drawing 1]

[Drawing 2]

[Drawing 3]

[Drawing 5]

[Drawing 6]

[Translation done.]

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.